Application No.: 10/698,820   Docket No.: MWS-062

AMENDMENTS TO THE SPECIFICATION

On page 1, please replace the paragraph in the "Related Applications" section with the following paragraph:

The illustrative embodiment of the present invention is related to a United States Patent Application entitled "Generating Code For Data References", United States Patent Application No. 09/876,487, the contents of which are hereby incorporated by reference.

On page 1, please replace the first paragraph in the "Background" section with the following paragraph:

Automatic code generation is a process whereby software source code is automatically produced from a graphical model. The software source code produced by the automatic code generation process may be compiled and then executed on a digital computer or other electronic device implementing the functionality specified by the model. The graphical model being studied may contain many different types of data references. Data may be used to represent the states of the system, as well as the flow of matter, energy, or information between system components. Each item of data in the model is defined to have a data storage class. Data is in turn represented in the generated software source code in a manner that is prescribed by its storage class. The software source code references data in a number of different ways including defining data, declaring data, initializing data, reading a value of data, assigning the value of data, and the choice of storage class controls how each of these references are generated.[[.]]

On page 3, please replace the third and fourth paragraphs starting at line 25 in the "Brief Description of the Drawings" section with the following paragraphs:

Figure 3 is a flowchart of the sequence of steps followed by the illustrative embodiment of the present invention to adjust and re-display source code generated following a change in the selected parameters for a custom storage class; and

Figure 4 depicts a view of the user interface of the illustrative embodiment of the present invention.

2

On page 5, please replace the paragraph starting at line 1 with the following paragraph:

The electronic device 2 is interfaced with a display 10. The display 10 includes a user interface 12 generated by the modeling and execution environment 4. The user interface 12, which is accessible by a user 30 through an input device such as a mouse, includes user-selectable parameter settings 14, 16 and 18. The user selectable parameter settings 14, 16 and 18 are parameters for use in the creation of a custom storage class 8 and are discussed in more detail below. The user-selectable parameter settings 14, 16 and 18 may be represented on the user interface 12 in a number of ways including through the use of conventional user interface controls such as radio buttons, check boxes or text boxes able to accept textual input specifying a parameter value. The user-selectable parameters are utilized by the automatic code generator 7 to create a custom storage class. The custom storage class is then utilized by the automatic code generator the create source code referencing data for the graphical model 6. The user interface 12 also includes a code view area 20 where the automatic code generator 7 may display salient aspect of the source code created from the use of the custom storage class.

On page 6, please replace the paragraph starting at line 19 with the following paragraph:

Custom data storage classes are not inherent to the modeling system and are generated by an interpreted programming language process. Using the interpreted programming language process, the user specifies a custom data storage class by specifying a set of instructions for each type of reference to the data. This set defines the data storage class. Specifically, the custom data storage class is characterized by the set of instructions defined by the user via parameter settings that detail how to generate software source code for each type of reference to data that is of that class. The parameter settings for the custom storage class are chosen by the user via the user interface 12. The user may then specify that custom data storage class as the item's storage class. Because the user specifies and defines these instructions, the possible variations in the software source code to be generated are extensive. The instructions corresponding to a given custom data storage class may be a function of the set of items that are defined to be of that class, so that the mechanism is self-referential.

3

Application No.: 10/698,820   Docket No.: MWS-062

On page 8, please replace the paragraph starting at line 12 with the following paragraph:

The initial generation of code responsive to the parameter settings may not always result in a desired outcome for the user. The illustrative embodiment of the present invention allows the process to iterate with a real time display of adjusted code reflective of each code recalculation. **Figure 3** is a flowchart of the sequence of steps followed by the illustrative embodiment of the present invention to adjust and re-display source code generated following a change in the selected parameters for a custom storage class. The sequence begins when the user selects parameters in the user interface 12 and the custom storage class is created (step 60). The custom storage class is then used to create source code referencing model data, and the source code is displayed to the user 30 in the code view 20 portion of the user interface 12 (step 62). Subsequently, the user 30 may change parameter settings and the process creates new source code (step 64). The new source code based on the adjusted parameter settings is then displayed to the user 30 (step 66).

On page 8, please replace the paragraph starting at line 26 with the following paragraph: **Figure 4** depicts a view 80 of the user interface 12 of the illustrative embodiment of the present invention. The view 80 includes a listing of different types of user selectable custom storage classes 82 (BitField, Const, ConstVolatile, Define, ExportToFile, ImportFromFile, Struct). Each type of custom storage class has multiple user-selectable parameter settings. For example the view 80 displays the general parameter settings for the Define storage class including a visibility parameter setting[[84]] 85, a memory access setting[[86]] 87, an initialization parameter setting[[88]] 89, a constant parameter setting[[90]] 91, a volatile parameter setting[[92]] 93 and a qualifier parameter setting[[94]] 95. Those skilled in the art will notice that the parameter settings displayed on the user interface include pull down menus 84, 86 and 88, check boxes 90 and 92 and text boxes 94 through which a user may specify parameter settings. The view 80 also includes a validation result area 102 and a code preview area 100. The validation result displays the result of a check to make sure selected parameter setting may be implemented without [[by]]the automatic code generator 7. For example, some parameter settings may conflict and prevent code generation. The code preview area is the area of the user interface where the automatic code generator displays generated source code and symbolic representations of source code.

4